Manual

CANoe VTP Protocol

Version 1.0.0

English

Imprint

Vector Informatik GmbH
Ingersheimer Str. 24

D-70499 Stuttgart

The information and data given in this user manual can be changed without prior notice. No part of this manual may be reproduced in any form or by any means without the written permission of the publisher, regardless of which method or which instruments, electronic or mechanical, are used. All technical information, images, drawings, etc. are protected by copyright law.
© Copyright 2011, Vector Informatik GmbH. Printed in Germany.
All rights reserved.

Table of Contents

1 Introduction 3

1.1 VTP at a Glance 4

1.2 History VTP Protocol 4

1.3 About this User Manual 4
- 1.3.1 Navigational Aids and Conventions 4
- 1.3.2 Certification 5
- 1.3.3 Warranty 5
- 1.3.4 Support 6
- 1.3.5 Registered Trademarks 6

2 General 7

2.1 Implementation Architecture 8
- 2.1.1 Overview 8
- 2.1.2 Radio Device Software 8
- 2.1.3 Personal Computer Software 8

2.2 Protocol Properties 9
- 2.2.1 UDP as Transport Foundation 9
- 2.2.2 Data Types 9
- 2.2.3 Radio Frame 10

2.3 Protocol Description 11
- 2.3.1 Structure of the Datagram 11
- 2.3.2 Datagram Header 11
- 2.3.3 Command Field 13

2.4 Protocol Commands 13
- 2.4.1 NOP Command 13
- 2.4.2 Start Command 13
- 2.4.3 Stop Command 13
- 2.4.4 Heartbeat Command 14
- 2.4.5 Statistic Command 14
- 2.4.6 TimeAdjustment Command 15
- 2.4.7 ScanRequest Command 15
- 2.4.8 ScanResponse Command 15
- 2.4.9 HardwareTypeRequest Command 16
- 2.4.10 HardwareTypeResponse Command 16
- 2.4.11 SoftwareVersionRequest Command 17
- 2.4.12 SoftwareVersionResponse Command 17
- 2.4.13 RadioMAC-ID-Request Command 17
- 2.4.14 RadioMAC-ID-Response Command 18
- 2.4.15 ChannelParameterRequest Command 18
- 2.4.16 ChannelParameterSet Command 18
- 2.4.17 ChannelParameterResponse Command 19
- 2.4.18 SelectRequest Command 19
- 2.4.19 SelectResponse Command 20
- 2.4.20 FreeSelection Command 20
- 2.4.21 AssignIFaceRequest Command 20
- 2.4.22 AssignIFaceResponse Command 20
- 2.4.23 TransmitFrame Command 21
- 2.4.24 ReceiveFrame Command 21

2.4.25 Reboot Command 22
2.4.26 Halt Command 22
2.4.27 Error Message 22

3 Appendix A: Address Table 25

1 Introduction

This chapter contains the following information:

1.1	VTP at a Glance	page 4
1.2	History VTP Protocol	page 4
1.3	About this User Manual	page 4
	Navigational Aids and Conventions	
	Certification	
	Warranty	
	Support	
	Registered Trademarks	

1.1 VTP at a Glance

IEEE 802.11p radio device access

CANoe VTP (**V**ector **T**ransport **P**rotocol) is a protocol for simple, fast, real-time exchange of IEEE 802.11p WLAN frames between CANoe and different radio devices via an Ethernet connection. The protocol is derived from the CANoe FDX (**F**ast **D**ata e**X**change) protocol and enables CANoe to use common CCUs (**C**ar **C**ommunication **U**nits) to receive and transmit WLAN traffic.

The WLAN frames are standardized in the reference documents
- IEEE 802.11:2007 and
- ETSI ES 202 663 V1.1.0 (2010-01).

1.2 History VTP Protocol

CANoe	Description
CANoe 7.6	First release of VTP feature
CANoe 7.6 SP2	User description to identify radio device added

1.3 About this User Manual

1.3.1 Navigational Aids and Conventions

To find information quickly

This user manual provides you with the following navigational aids:

> At the beginning of each chapter you will find a summary of the contents
> The header shows which chapter and paragraph you are located in
> The footer shows which version the user manual refers to

Conventions

The following two charts show the spelling and symbol conventions used in this manual.

Style	Utilization
bold	Fields, interface elements, window and dialog names in the software. Accentuation of warnings and notes. **[OK]** Buttons are denoted by square brackets **File** | **Save** Notation for menus and menu entries
CANoe	Legally protected proper names and side notes.
`Source code`	File name and source code.
Hyperlink	Hyperlinks and references.
<Ctrl>+<S>	Notation for keyboard shortcuts.

 © Vector Informatik GmbH

Symbol	Utilization
	Here you can obtain supplemental information.
	This symbol calls your attention to warnings.
	Here you can find additional information.
	Here is an example that has been prepared for you.
	Step-by-step instructions provide assistance at these points.
	Instructions on editing files are found at these points.
	This symbol warns you not to edit the specified file.
	This symbol indicates multimedia files like e.g. video clips.
	This symbol indicates an introduction into a specific topic.
	This symbol indicates text areas containing basic knowledge.
	This symbol indicates text areas containing expert knowledge.
	This symbol indicates that something has changed.

1.3.2 Certification

Certified Quality Management System

Vector Informatik GmbH has ISO 9001:2008 certification. The ISO standard is a globally recognized quality standard.

1.3.3 Warranty

Limitation of warranty

We reserve the right to change the contents of the documentation and the software without notice. Vector Informatik GmbH assumes no liability for correct contents or damages which are resulted from the usage of the user manual. We are always grateful for references to mistakes or for suggestions for improvement, so as to be able to offer you even better-performing products in the future.

1.3.4 Support

Need support?

You can get through to our hotline by calling

+49 (711) 80670-200

or you can send a problem report to the Vector Support.

1.3.5 Registered Trademarks

Registered trademarks

All trademarks mentioned in this user manual, including those registered to third parties, are governed by the respective trademark laws and are the property of their respective owners. All trademarks, trade names or company names are or can be trademarks or registered trademarks of their particular owners. All rights which are not expressly allowed are reserved. Failure to explicitly note any given trademark within this user manual does not imply that a third party does not have rights to it.

> Windows, Windows XP, Vista, Windows 7 are trademarks of the Microsoft Corporation.

> CANoe is a trademark of Vector Informatik GmbH.

2 General

This chapter contains the following information:

2.1	Implementation Architecture	page 8
	Overview	
	Radio Device Software	
	Personal Computer Software	
2.2	Protocol Properties	page 9
	UDP as Transport Foundation	
	Data Types	
	Radio Frame	
2.3	Protocol Description	page 11
	Structure of the Datagram	
	Datagram Header	
	Command Field	
2.4	Protocol Commands	page 13
	NOP Command	
	Start Command	
	Stop Command	
	Heartbeat Command	
	Statistic Command	
	TimeAdjustment Command	
	ScanRequest Command	
	ScanResponse Command	
	HardwareTypeRequest Command	
	HardwareTypeResponse Command	
	SoftwareVersionRequest Command	
	SoftwareVersionResponse Command	
	RadioMAC-ID-Request Command	
	RadioMAC-ID-Response Command	
	ChannelParameterRequest Command	
	ChannelParameterSet Command	
	ChannelParameterResponse Command	
	SelectRequest Command	
	SelectResponse Command	
	FreeSelection Command	
	AssignIFaceRequest Command	
	AssignIFaceResponse Command	
	TransmitFrame Command	
	ReceiveFrame Command	
	Reboot Command	
	Halt Command	
	Error Message	

2.1 Implementation Architecture

2.1.1 Overview

System Design

To get CANoe talking IEEE 802.11p it needs to have access to a radio device for transmission and reception of WLAN frames. This radio device is connected to the PC running CANoe via Ethernet IEEE 802.3. The communication protocol is based on TCP/IP using the user datagram protocol (UDP) to exchange commands with VTP.

The VTP protocol is handled by different software implementations on the PC and the radio device, for a schematic see Figure 1.

2.1.2 Radio Device Software

RMTPCD

The remote service software of the radio device is implemented as a background process (daemon) running on a Linux operating system. The name is built by the acronym RMTPCD, which stands for **R**e**m**o**t**e **P**acket **C**apture **D**aemon. To get a good communication responsiveness of the remote service software, the process is divided into several threads (POSIX). The connection to CANoe is established via UDP sockets.

Unassigned radio devices do listen by default on port number 49856 (0xC2C0), until they are selected by CANoe to get an additional port number e.g. 49857 (0xC2C1) for the exchange of WLAN frames. RMTPCD opens a raw socket in promiscuous mode to the IEEE 802.11p interface of the radio device to receive and transmit WLAN frames. Hint: The monitoring mode of the IEEE 802.11p interface offers a much more detailed analysis of the WLAN traffic, but does not allow transmitting frames, only receiving them. Therefore this mode is not used by RMTPCD.

2.1.3 Personal Computer Software

CANoe DLL

The implementation of the communication software on the PC is developed as DLL and part of CANoe. The connection to the radio device is established via UDP sockets and starts with using port number 49856 (0xC2C0). After selection of a radio device CANoe does assign an additional port number e.g. 49857 (0xC2C1) and following numbers according to the number of radio devices already collected.

Every VTP communication is initiated by CANoe only.

 © Vector Informatik GmbH

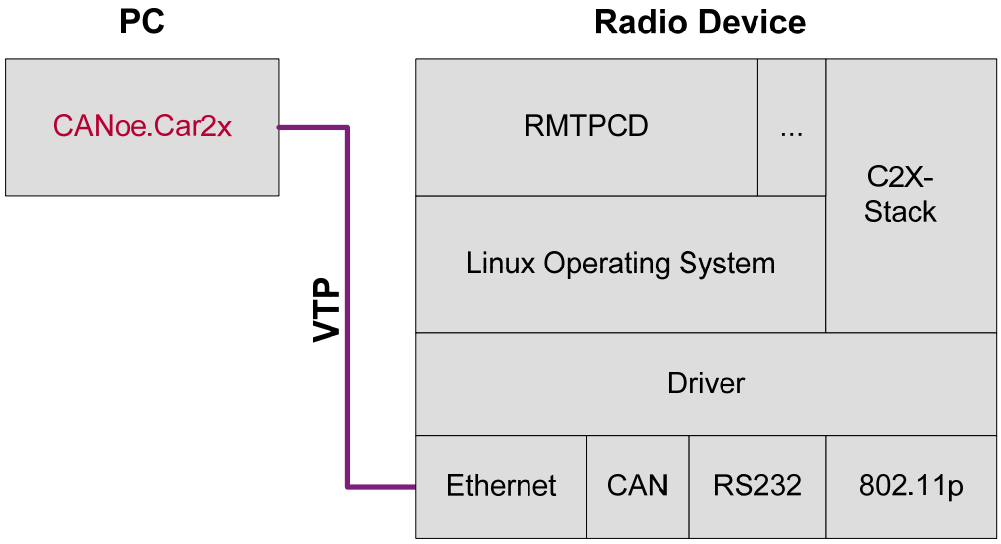

Figure 1: System Design

2.2 Protocol Properties

2.2.1 UDP as Transport Foundation

UDP protocol

The protocol is based on the UDP protocol (IPv4). UDP is a widespread standard protocol, which means there will most likely be an implementation available on the radio device.

The exchange of data between CANoe and the radio device is achieved through reciprocal transmission of UDP datagrams. As a matter of principle, CANoe always sends a datagram to the radio device first and therefore has to know the IP address used by the radio device for the VTP protocol. CANoe may scan every attached network to determine the available radio devices and their IP addresses.

UDP port

By default, CANoe uses port 0xC2C0 for the VTP protocol. The port number is adjustable on the radio device and in the CANoe WLAN settings.

The exact structure of the UDP datagrams is shown in section 2.3 Protocol Description. The datagram structures have been kept purposely simple so that an experienced developer can implement the VTP protocol within a short time on the radio device provided there is already an IP stack available.

2.2.2 Data Types

Data types

To reduce the bandwidth required for data transfer over the network, the VTP protocol is binary coded. To facilitate the data access on the radio device side, the data types are limited to these ones that are directly available in most C/C++ compilers and most other programming languages.

These data types are

- int8 (1 Byte signed integer)
- uint8 (1 Byte unsigned integer)
- int16 (2 Byte signed integer)
- uint16 (2 Byte unsigned integer)
- int32 (4 Byte signed integer)
- uint32 (4 Byte unsigned integer)
- int48 (6 Byte signed integer)
- uint48 (6 Byte unsigned integer)
- int64 (8 Byte signed integer)
- uint64 (8 Byte unsigned integer)
- float (4 Byte floating point)
- double (8 Byte floating point)
- string (null terminated ASCII character string)
- bytearray (sequence of individual data bytes)

Encoding

The data types are encoded for the network architecture (integer values use Big Endian as the byte sequence, signed integer types use the 2's complement, and floating-point values use IEEE format).

Strings are made up of single-byte ASCII characters and are always null-byte terminated. This corresponds to the string format from the C programming language.

Bytearrays go along with a length value indicating the number of data bytes used in the array.

Offset

For the definition of data sets each set is assigned a fixed offset (value in bytes). You have to ensure that data sets do not overlap in the memory layout.

2.2.3 Radio Frame

IEEE 802.11p frame

The **IEEE 802.11p** radio frame consists of the MAC header, the payload (frame body) and the checksum (FCS). If the radio device cannot provide some values, they are set to zero. VTP inserts an Ethernet header (IEEE 802.3) between the MAC header and the frame body.

Offset	Size	Type	Field	Description
0	2	uint16	frameControl	Frame Control
2	2	uint16	commandCode	Duration/ID
4	6	uint48	addressOne	Address 1 (Destination)
10	6	uint48	addressTwo	Address 2 (Source)
16	6	uint48	addressThree	Address 3 (BSSID)
22	2	uint16	sequenceControl	Sequence Control
24	6	uint48	addressFour	Address 4 (Relay)
30	2	uint16	qosControl	Quality of Service Control
32	6	uint48	ethDestAddress	Ethernet Destination Address (MAC Header DA)

 © Vector Informatik GmbH

Offset	Size	Type	Field	Description
38	6	uint48	ethSrcAddress	Ethernet Source Address (MAC Header SA)
44	2	uint16	ethType	Ethernet Type
46	n	…	frameBody	Frame Body (0..2314)
…	4	uint32	frameChecksum	FCS (MAC Header frame checksum)

2.3 Protocol Description

2.3.1 Structure of the Datagram

Communication via datagrams

The communication between CANoe and the radio device is implemented through datagrams that are exchanged via UDP. A datagram consists of the datagram header followed by one or more commands.

Offset	Size	Field	Description
0	16	datagramHeader	Header
16	aa	commandField1	First command
16 + aa	*bb*	*commandField2*	*Optional second command*
16 + aa + bb	*cc*	*commandField3*	*Optional third commands*
…	…	…	*Optional further commands*

The datagram header and the individual commands are described in the following sections.

2.3.2 Datagram Header

Datagram header

The datagram header consists of a signature, a two-digit version number (major and minor version) for the protocol, the number of subsequent commands and a sequence number. For future development of VTP protocol extensions and for better alignment two more bytes are reserved.

Offset	Size	Type	Field	Description
0	8	uint64	vtpSignature	Signature of the VTP protocol, this is always the value 0x505456656F4E4143
8	1	uint8	vtpMajorVersion	Protocol version (major part) kVtpMajorVersion = 1
9	1	uint8	vtpMinorVersion	Protocol version (minor part) kVtpMinorVersion = 0
10	2	uint16	numberOfCommands	Number of commands in the datagram
12	2	uint16	sequenceNumber	Datagram sequence number
14	2	uint16	reserved	This field should be initialized with 0.

Signature

The signature serves as a magic cookie to check whether the datagram was actually intended as VTP communication (or whether a program sent the datagram to CANoe or a radio device by mistake). CANoe and the radio device ignore all datagrams that have a wrong signature.

Version number

The two-digit version number is used to check compatibility. A new protocol version with the same major version number may contain additional information but the datagram can still be processed, except for the new information, by older programs with the same major version number. The protocol remains compatible until the major version number changes.

Sequence number

The sequence number enables you to recognize the loss of individual datagrams. The datagrams are numbered sequentially from 0x0001 to 0x7FFF by the sender. The special value 0x0000 is used to start a new counting sequence. To end a counting sequence the current sequence number is ORed with the value 0x8000. The overwrap of sequence numbers goes from 0x7FFF auf 0x0001.

Example

Sequence number	Description
0x0000	Special number 0x0000 indicates start of sequence counting.
0x0001	
0x0002	
0x0003	
…	
0x7FFF	Overwrap of sequence number happens from value 0x7FFF to value 0x0001
0x0001	
0x0002	
…	
0x1233	
0x1234	
0x9235	Sequence number 0x1235 is combined with 0x8000 to indicate the end of sequence counting.

Sequence counting

Sequence counting is an optional feature. Use of the special value 0x8000 as sequence number indicates that no sequence counting is being used. Both transmission directions (CANoe to radio device, radio device to CANoe) have their dedicated sequence counter.

When a datagram is received that indicates the same sequence number in the range from 0x0001 to 0x7FFF as its predecessor, it will be ignored.

If CANoe determines any sequence numbering errors in the incoming datagrams, a warning is written to the CANoe Write window.

 © Vector Informatik GmbH

2.3.3 Command Field

Command Field

Every command begins with the **CommandCode** and the size of the command. By specifying the size of the command it is possible to insert additional fields later without this resulting in incompatibility with the protocol. The **CommandCode** determines what type of command is being used (measurement start, frame transmission, etc.).

Command Size

The command size refers to the size of the whole command field including the size field itself. Therefore the size can never be less than four. A value of zero or an odd value will lead to refuse the remaining commands of this datagram.

Offset	Size	Type	Field	Description
0	2	unit16	commandSize	Size of this command in bytes
2	2	unit16	commandCode	Command code
4	*…*	*…*	*…*	*Optional further fields*

The individual commands are described in the following section.

2.4 Protocol Commands

2.4.1 NOP Command

No operation

The **NOP** (**N**o **op**eration) command does nothing. No action will be performed. The reason for the existents of this command is just to get the protocol more robust against a zero copying failure into the datagram.

Offset	Size	Type	Field	Description
0	2	uint16	commandSize	Size of command (4 Bytes)
2	2	uint16	commandCode	kCommandCode_Nop = 0x0000

2.4.2 Start Command

Start a measurement

The **Start** command is sent to the radio device by CANoe in order to start a measurement. The radio device begins to transmit all received IEEE 802.11p frames to CANoe according to its configuration. If the measurement is already running the **Start** command is ignored.

Offset	Size	Type	Field	Description
0	2	uint16	commandSize	Size of command (4 Bytes)
2	2	uint16	commandCode	kCommandCode_Start = 0x0001

2.4.3 Stop Command

Stop a measurement

The **Stop** command is sent to the radio device by CANoe in order to stop a measurement. The radio device ends to transmit all received IEEE 802.11p frames to CANoe. If the measurement is not running the **Stop** command is ignored.

Offset	Size	Type	Field	Description
0	2	uint16	commandSize	Size of command (4 Bytes)
2	2	uint16	commandCode	kCommandCode_Stop = 0x0002

2.4.4 Heartbeat Command

Heartbeat

The **Heartbeat** command is sent to the radio device by CANoe cyclical. It has to reach the radio device before expiration of the heartbeat time of about two seconds otherwise the radio device will switch to measurement stop by itself. With every received heartbeat command the radio device starts a new countdown and transmits its statistic information to CANoe.

Offset	Size	Type	Field	Description
0	2	uint16	commandSize	Size of command (4 Bytes)
2	2	uint16	commandCode	kCommandCode_Heartbeat = 0x0003

2.4.5 Statistic Command

Bus statistic

The **Statistic** command is sent to CANoe by the radio device to respond to the heartbeat command. The timestamp is given in UNIX time format (time zone set to UTC). The channel values are accumulated in each heartbeat interval and reset to zero after transmission of the statistic command.

Offset	Size	Type	Field	Description
0	2	uint16	commandSize	Size of command (at least 14 Bytes)
2	2	uint16	commandCode	kCommandCode_Statistic = 0x0004
4	4	uint32	timeSeconds	Seconds
8	4	uint32	timeMicroSeconds	Microseconds
12	1	uint8	numberOfInterfaces	Number of WLAN interfaces
13	1	uint8	reserved	Reserved byte
14	2	uint16	numberOfFramesRxCh1	Channel 1: Received frames
16	4	uint32	numberOfBytesRxCh1	Channel 1: Received bytes
20	2	uint16	numberOfFramesTxCh1	Channel 1: Transmitted frames
22	4	uint32	numberOfBytesTxCh1	Channel 1: Transmitted bytes
26	2	uint16	numberOfCollisionsCh1	Channel 1: Collisions
28	2	uint16	numberOfErrorsCh1	Channel 1: Errors
30	…	…	…	Further channels

 © Vector Informatik GmbH

2.4.6 TimeAdjustment Command

Time adjustment

The **TimeAdjustment** command is sent to the radio device by CANoe in order to provide the PC time to the radio device for time adjustment. This command is sent by CANoe before start of measurement. The radio device evaluates the time difference between its internal clock and the PC time. This difference time is added to the internal timestamp of each IEEE 802.11p frame to build the correct CANoe time. The timestamp is given in UNIX time format (time zone set to UTC).

Offset	Size	Type	Field	Description
0	2	uint16	commandSize	Size of command (12 Bytes)
2	2	uint16	commandCode	kCommandCode_SetTime = 0x0005
4	4	uint32	timeSeconds	Seconds
8	4	uint32	timeMicroSeconds	Microseconds

2.4.7 ScanRequest Command

Device scan request

The **ScanRequest** command is broadcasted on port 0xC2C0 to the cable-connected network by CANoe to gather information about all available radio devices. All attached radio devices will answer with a **ScanResponse** command to show their status.

Offset	Size	Type	Field	Description
0	2	uint16	commandSize	Size of command (4 Bytes)
2	2	uint16	commandCode	kCommandCode_ScanRequest = 0x0008

2.4.8 ScanResponse Command

Device scan response

The **ScanResponse** command is sent to CANoe by the radio device. This command is the answer to a device scan requested by CANoe and provides information about the hardware type and number of WLAN channels. The radio device indicates if it is already assigned by another CANoe and not available for the requesting one.

Offset	Size	Type	Field	Description
0	2	uint16	commandSize	Size of command (10 Bytes + length of identification string)
2	2	uint16	commandCode	kCommandCode_ScanResponse = 0x0009
4	1	uint8	hardwareType	Hardware type: - unknown (0) - NEC LinkBird-MX 2 (1) - Denso WSU-01 (2) - NEC LinkBird-MX 3 (3) - Delphi CRPU-M2 (4) - Cohda MK2 (5) - Denso WSU-02 (6)

Offset	Size	Type	Field	Description
				- Delphi CRPU-M3 (7) - Cohda MK3 (8)
5	1	uint8	deviceStatus	Device status = 0x01: device in use, 0x02: device not configurable
6	1	uint8	numberOfInterfaces	Number of WLAN interfaces
7	1	uint8	reserved	Reserved byte
8	2	uint16	identStringLength	Length of the identification string
10	n	uint16	identificationString	Identification string

2.4.9 HardwareTypeRequest Command

Hardware type request

The **HardwareTypeRequest** command is sent to the radio device by CANoe to request the hardware type. The radio device will answer with a **HardwareTypeResponse** command to provide information about the hardware type.

Offset	Size	Type	Field	Description
0	2	uint16	commandSize	Size of command (4 Bytes)
2	2	uint16	commandCode	kCommandCode_HardwareTypeRequest = 0x000A

2.4.10 HardwareTypeResponse Command

Hardware type response

The **HardwareTypeResponse** command is sent to CANoe by the radio device. This command is the answer to a hardware type request by CANoe and provides information about the hardware type and number of WLAN channels.

Offset	Size	Type	Field	Description
0	2	uint16	commandSize	Size of command (at least 12 Bytes)
2	2	uint16	commandCode	kCommandCode_HardwareTypeResponse = 0x000B
4	1	uint8	hardwareType	Hardware type: - unknown (0) - NEC LinkBird-MX 2 (1) - Denso WSU-01 (2) - NEC LinkBird-MX 3 (3) - Delphi CRPU-M2 (4) - Cohda MK2 (5) - Denso WSU-02 (6) - Delphi CRPU-M3 (7) - Cohda MK3 (8)
5	1	uint8	numberOfInterfaces	Number of WLAN interfaces

 © Vector Informatik GmbH

Offset	Size	Type	Field	Description
6	2	uint16	capabilityBits	Radio device feature list = 0x01: genuine MAC-header, 0x02: RX/TX-counter, 0x04: error counter, 0x08: checksum (FCS), 0x10: accurate time (e.g. GPS, DCF77)
8	4 * n	uint32	vendorIdWlan	Radio Chip Manufacturer

2.4.11 SoftwareVersionRequest Command

Software version request

The **SoftwareVersionRequest** command is sent to the radio device by CANoe to request the software version. The radio device will answer with a **SoftwareVersionResponse** command to provide information about the software version.

Offset	Size	Type	Field	Description
0	2	uint16	commandSize	Size of command (4 Bytes)
2	2	uint16	commandCode	kCommandCode_SoftwareVersionRequest = 0x000C

2.4.12 SoftwareVersionResponse Command

Software version response

The **SoftwareVersionResponse** command is sent to CANoe by the radio device. This command is the answer to a software version request by CANoe and provides information about the software version.

Offset	Size	Type	Field	Description
0	2	uint16	commandSize	Size of command (8 Bytes)
2	2	uint16	commandCode	kCommandCode_SoftwareVersionResponse = 0x000D
4	1	uint8	softwareMainVersion	Software main version
5	1	uint8	softwareSubVersion	Software sub version
6	1	uint8	softwareReleaseVersion	Software release version
7	1	uint8	reserved	Reserved byte

2.4.13 RadioMAC-ID-Request Command

Radio MAC-ID request

The **RadioMAC-ID-Request** command is sent to the radio device by CANoe to request the hardware MAC-ID of the radio device on the chosen channel. The radio device will answer with a **RadioMAC-ID-Response** command to provide the MAC-ID.

Offset	Size	Type	Field	Description
0	2	uint16	commandSize	Size of command (6 Bytes)
2	2	uint16	commandCode	kCommandCode_RadioMAC-ID-Request = 0x000E
4	1	uint8	interfaceNumber	Number of WLAN interface
5	1	uint8	reserved	Reserved byte

2.4.14 RadioMAC-ID-Response Command

Radio MAC-ID response

The **RadioMAC-ID-Response** command is sent to CANoe by the radio device. This command is the answer to a radio MAC-ID request by CANoe and provides the hardware MAC-ID of the radio device on the chosen channel.

Offset	Size	Type	Field	Description
0	2	uint16	commandSize	Size of command (12 Bytes)
2	2	uint16	commandCode	kCommandCode_RadioMAC-ID-Response = 0x000F
4	1	uint8	interfaceNumber	Number of WLAN interface
5	1	uint8	reserved	Reserved byte
6	6	uint48	radioMacId	Radio-MAC-ID

2.4.15 ChannelParameterRequest Command

Channel parameter request

The **ChannelParameterRequest** command is sent to the radio device by CANoe to request the channel parameters of the radio device on the chosen channel. The radio device will answer with a **ChannelParameterResponse** command to provide the channel information.

Offset	Size	Type	Field	Description
0	2	uint16	commandSize	Size of command (6 Bytes)
2	2	uint16	commandCode	kCommandCode_ChannelParameterRequest = 0x0010
4	1	uint8	interfaceNumber	Number of WLAN interface
5	1	uint8	reserved	Reserved byte

2.4.16 ChannelParameterSet Command

Channel parameter set

The **ChannelParameterSet** command is sent to the radio device by CANoe to set the channel parameters of the radio device on the chosen channel. The radio device will answer with a **ChannelParameterResponse** command to provide the current settings. If some of the requested parameters are not changeable, the old values will remain. The measurement units are declared in IEEE 802.11a.

Offset	Size	Type	Field	Description
0	2	uint16	commandSize	Size of command (10 Bytes)
2	2	uint16	commandCode	kCommandCode_ChannelParameterSet

 © Vector Informatik GmbH

Offset	Size	Type	Field	Description
				= 0x0011
4	1	uint8	interfaceNumber	Number of WLAN interface
5	1	uint8	channelNumber	Channel number
6	1	uint8	bandWidth	Bandwidth (MHz)
7	1	uint8	bitRate	Bitrate (Mbit/s)
8	1	uint8	transmissionPower	Transmission power (dBm)
9	1	uint8	reserved	Reserved byte

2.4.17 ChannelParameterResponse Command

Channel parameter response

The **ChannelParameterResponse** command is sent to CANoe by the radio device. This command is the answer to a channel parameter request or channel parameter set command by CANoe and provides the current settings of the radio device on the chosen channel.

Offset	Size	Type	Field	Description
0	2	uint16	commandSize	Size of command (10 Bytes)
2	2	uint16	commandCode	kCommandCode_ChannelParameterResponse = 0x0012
4	1	uint8	interfaceNumber	Number of WLAN interface
5	1	uint8	channelNumber	Channel number
6	1	uint8	bandWidth	Bandwidth (MHz)
7	1	uint8	bitRate	Bitrate (Mbit/s)
8	1	uint8	transmissionPower	Transmission power (dBm)
9	1	uint8	reserved	Reserved byte

2.4.18 SelectRequest Command

Device selection request

The **SelectRequest** command is sent to the radio device by CANoe to allocate the radio device for exclusive communication with this CANoe instance. A new TCP/IP port is assigned for all further VTP communication (recommended use of 0xC2C1 and the following, according to the connection number of the radio device). The radio device will answer with a **SelectResponse** command already on the new port to accept the connection with CANoe.

Offset	Size	Type	Field	Description
0	2	uint16	commandSize	Size of command (6 Bytes)
2	2	uint16	commandCode	kCommandCode_SelectRequest = 0x0013
4	2	uint16	portNumber	TCP/IP port number

2.4.19 SelectResponse Command

Device selection response

The **SelectResponse** command is sent to CANoe by the radio device. This command is the positive answer to a selection request by CANoe. The radio device is now occupied by the requesting CANoe instance and its status is ‘device in use’ for scan requests of other instances. The occupation will be cancelled by a free selection command or the occurrence of a heartbeat event. The sequence counter is reset.

Offset	Size	Type	Field	Description
0	2	uint16	commandSize	Size of command (4 Bytes)
2	2	uint16	commandCode	kCommandCode_SelectResponse = 0x0014

2.4.20 FreeSelection Command

Free device selection

The **FreeSelection** command is sent to the radio device by CANoe to cancel its allocation of the radio device. The status ‘device in use’ is cleared.

Offset	Size	Type	Field	Description
0	2	uint16	commandSize	Size of command (4 Bytes)
2	2	uint16	commandCode	kCommandCode_FreeSelection = 0x0015

2.4.21 AssignIFaceRequest Command

Interface assignment request

The **AssignIFaceRequest** command is sent to the radio device by CANoe to allocate or deallocate a WLAN interface of the radio device. The radio device will answer with an **AssignIFaceResponse** command.

Offset	Size	Type	Field	Description
0	2	uint16	commandSize	Size of command (6 Bytes)
2	2	uint16	commandCode	kCommandCode_AssignIFaceRequest = 0x0016
4	1	uint8	interfaceNumber	Number of WLAN interface
5	1	uint8	selectionFlag	Interface assignment = 0x01: allocate device, 0x00: deallocate device

2.4.22 AssignIFaceResponse Command

Interface assignment response

The **AssignIFaceResponse** command is sent to CANoe by the radio device. This command is the answer to an assign interface request by CANoe. The radio device indicates the current assignment state of the chosen WLAN interface.

Offset	Size	Type	Field	Description
0	2	uint16	commandSize	Size of command (4 Bytes)
2	2	uint16	commandCode	kCommandCode_AssignIFaceResponse

 © Vector Informatik GmbH

Offset	Size	Type	Field	Description
				= 0x0017
4	1	uint8	interfaceNumber	Number of WLAN interface
5	1	uint8	selectionFlag	Interface assignment = 0x01: device allocated, 0x00: device deallocated

2.4.23 TransmitFrame Command

Radio frame transmission

The **TransmitFrame** command is sent to the radio device by CANoe. This command triggers the transmission of the included IEEE 802.11p frame on the chosen interface.

Offset	Size	Type	Field	Description
0	2	uint16	commandSize	Size of command (10 Bytes + length of WLAN frame)
2	2	uint16	commandCode	kCommandCode_TransmitFrame = 0x001E
4	1	uint8	interfaceNumber	Number of WLAN interface
5	1	uint8	channelNumber	Number of WLAN channel
6	2	uint16	transactionId	Transaction identifier
8	2	uint16	frameLength	Frame length
10	n	…	wlanFrame	IEEE 802.11p WLAN frame

2.4.24 ReceiveFrame Command

Radio frame reception

The **ReceiveFrame** command is sent to CANoe by the radio device. This command includes an IEEE 802.11p frame received on the indicated interface. Every frame transmitted on the indicated interface is signalled to CANoe with direction set to TX. The timestamp is given in UNIX time format (time zone set to UTC).

Offset	Size	Type	Field	Description
0	2	uint16	commandSize	Size of command (24 Bytes + length of WLAN frame)
2	2	uint16	commandCode	kCommandCode_ReceiveFrame = 0x001F
4	1	uint8	interfaceNumber	Number of WLAN interface
5	1	uint8	channelNumber	Number of WLAN channel
6	1	uint8	direction	Transmission direction = 0x01: TX, 0x00: RX
7	1	uint8	reserved	Reserved byte
8	4	uint32	timeSeconds	Seconds
12	4	uint32	timeMicroSeconds	Microseconds
16	2	uint16	signalStrength	Signal strength

Offset	Size	Type	Field	Description
18	2	uint16	signalQuality	Signal quality
20	2	uint16	transactionId	Transaction identifier
22	2	uint16	frameLength	Frame length
24	n	…	wlanFrame	IEEE 802.11p WLAN frame

2.4.25 Reboot Command

Reboot device

The **Reboot** command is sent to the radio device to bring the operating system to shut down the radio device and start again. This command is not initiated by CANoe, but may be used by automated scripts.

Offset	Size	Type	Field	Description
0	2	uint16	commandSize	Size of command (4 Bytes)
2	2	uint16	commandCode	kCommandCode_Reboot = 0x003D

2.4.26 Halt Command

Halt device

The **Halt** command is sent to the radio device to bring the operating system to shut down the radio device to power off. This command is not initiated by CANoe, but may be used by automated scripts.

Offset	Size	Type	Field	Description
0	2	uint16	commandSize	Size of command (4 Bytes)
2	2	uint16	commandCode	kCommandCode_Reboot = 0x003E

2.4.27 Error Message

Error status message

The **Error** message is sent by the radio device or by CANoe depending on the origin of the failure. If the error message is related to a command that cannot be processed, the sequence number of the rejected datagram will be the error source. If the error message is related to a WLAN interface, the interface number will be the error source. In any other case the error source is set to zero.

Offset	Size	Type	Field	Description
0	2	uint16	commandSize	Size of command (16 Bytes)
2	2	uint16	commandCode	kCommandCode_Error = 0x003F
4	2	uint16	errorCode	Error code
6	2	uint16	errorSource	Error source
8	8	uint64	errorDesc	Description

 © Vector Informatik GmbH

Error codes

The error codes with description.

Error Code	Description
1001: kErrorCode_QueueFull	Too many commands to process.
1002: kErrorCode_MissingFrame	Sequence numbering violation.
1003: kErrorCode_InterfaceOffline	WLAN interface not available.
1004: kErrorCode_WrongParameter	General parameter mismatch.
1005: kErrorCode_DeviceUnavailable	Negative response to selection command.
1006: kErrorCode_WrongChannel	Transmit frame channel number different to device settings.

3 Appendix A: Address Table

Vector Informatik GmbH	Vector Informatik GmbH Ingersheimer Str. 24 70499 Stuttgart Germany Phone: +49 711 80670-0 Fax: +49 711 80670-111 mailto:info@de.vector.com http://www.vector.com
Vector Informatik GmbH Shanghai Representative Office	Vector Informatik GmbH Shanghai Representative Office Unit 1701, Sunyoung Center No.398 Jiangsu Road Changning District Shanghai 200050 P.R. China Phone: +86 21 6432 53530 Fax: +86 21 6432 5308 mailto:info@cn.vector.com http://www.vector.com
Vector CANtech, Inc.	Vector CANtech, Inc. Suite 550 39500 Orchard Hill Place Novi, Mi 48375 USA Phone: +1 248 449 9290 Fax: +1 248 449 9704 mailto:info@us.vector.com http://www.vector.com

Vector France SAS
Vector France SAS
168, Boulevard Camélinat
92240 Malakoff
France
Phone: +33 1 4231 4000
Fax: +33 1 4231 4009
mailto:information@fr.vector.com
http://www.vector.com

Vector GB Ltd.
Vector GB Ltd.
Rhodium, Central Boulevard
Blythe Valley Park
Solihull, Birmingham
West Midlands, B90 8AS
United Kingdom
Phone: +44 121 50 681-50
Fax: +44 121 50 681-69
mailto:info@uk.vector.com
http://www.vector.com

Vector Informatik India Private Limited
Vector Informatik India Private Limited
4/1/1/1, 3rd floor, Sutar Icon
Sus Road
Pashan
Pune 411021
India
Phone: +91 20 2587 2023
Fax: +91 20 2587 2025
mailto:info@in.vector.com
http://www.vector.com

 © Vector Informatik GmbH

Vector Japan Co., Ltd.
Vector Japan Co., Ltd.
Seafort Square Center Bld. 18F
2-3-12, Higashi-shinagawa
Shinagawa-ku
Tokyo 140-0002
Japan
Phone: +81 3 5769 7800
Fax: +81 3 5769 6975
mailto:info@jp.vector.com
http://www.vector.com

Vector Korea IT Inc.
Vector Korea IT Inc.
1406 Mario Tower
Guro-dong, Guro-gu, 222-12
Seoul 152-848
Republic of Korea
Phone: +82 2 8070 600
Fax: +82 2 8070 601
mailto:info@kr.vector.com
http://www.vector.com

VecScan AB
VecScan AB
Theres Svenssons Gata 9
417 55 Göteborg
Sweden
Phone: +46 (31) 764 7600
Fax: +46 (31) 764 7619
mailto:info@se.vector.com
http://www.vector.com

Get more Information!

Visit our Website for:

> News

> Products

> Demo Software

> Support

> Training Classes

> Addresses

www.vector.com